

IN THE CLAIMS

Please amend the claims as follows.

1 1. (Currently Amended) An apparatus for computer hardware multithreading
2 comprising:
3 a plurality of processors, each processor having hardware support for the
4 capability of executing a plurality of threads;
5 a memory coupled to the plurality of processors; and
6 a thread dispatch mechanism residing in the memory and executed by at least one
7 of the plurality of processors, the thread dispatch mechanism determining which of the
8 plurality of processors are idle ~~when the processor is able to accept a new thread for each~~
9 ~~processor hardware threads~~, which of the plurality of processors is busy processing a
10 thread but can accept an additional thread, and which of the plurality of processors cannot
11 accept an additional thread since it is working on a maximum number of threads the
12 processor can execute and, the thread dispatch mechanism dispatching a new thread to an
13 idle processor, if one exists.

1 2. (Original) The apparatus of claim 1 wherein, if none of the plurality of
2 processors is idle and if at least one of the plurality of processors can accept an
3 additional thread, the thread dispatch mechanism dispatches the new thread to one
4 of the plurality of processors that can accept an additional thread.

1 3. (Original) The apparatus of claim 1 wherein, if all of the plurality of
2 processors cannot accept an additional thread, the thread dispatch mechanism
3 waits for one of the plurality of processors to complete processing a thread,
4 thereby becoming a processor that can accept an additional thread, and then
5 dispatches the thread to the processor that can accept an additional thread.

1 4. (Currently Amended) A method for dispatching threads in a computer system that
2 includes a plurality of processors that can each support hardware multithreading to
3 execute a plurality of threads, the method comprising the steps of:
4 (1) determining the status of each of the plurality of processors, wherein a
5 processor is idle if not executing any threads ~~when able to accept a new thread for each~~
6 ~~processor hardware thread~~, wherein the processor can accept an additional thread if busy
7 working on one or more threads but has the capacity to process the additional thread, and
8 wherein the processor cannot accept an additional thread if busy working on a maximum
9 number of threads the processor can execute; and
10 (2) dispatching a new thread to an idle processor, if one exists.

1 5. (Original) The method of claim 4 further comprising the step of:
2 if none of the plurality of processors is idle and if at least one of the plurality of
3 processors can accept an additional thread, the thread dispatch mechanism dispatches the
4 new thread to one of the plurality of processors that can accept an additional thread.

1 6. (Original) The method of claim 4 further comprising the steps of:
2 if all of the plurality of processors cannot accept an additional thread, the thread
3 dispatch mechanism waits for one of the plurality of processors to complete processing a
4 thread, thereby becoming a processor that can accept an additional thread, and then
5 dispatches the thread to the processor that can accept an additional thread.

1 7. (Currently Amended) A program product comprising:
2 (A) a thread dispatch mechanism that determines which of a plurality of
3 processors in a hardware multithreading, multiprocessor computer system are idle ~~when~~
4 ~~the processor is able to accept a new thread for each processor hardware thread,~~ which of
5 the plurality of processors is busy but can accept an additional thread, and which of the
6 plurality of processors cannot accept an additional thread since it is working on a
7 maximum number of threads the processor can execute, the thread dispatch mechanism
8 dispatching a new thread to an idle processor, if one exists, wherein each processor can
9 execute a plurality of threads; and
10 (B) computer-readable signal bearing media bearing the thread dispatch
11 mechanism.

1 8. (Original) The program product of claim 7 wherein the computer-readable
2 signal bearing media comprises recordable media.

1 9. (Original) The program product of claim 7 wherein the computer-readable
2 signal bearing media comprises transmission media.

1 10. (Original) The program product of claim 7 wherein, if none of the plurality of
2 processors is idle and if at least one of the plurality of processors can accept an
3 additional thread, the thread dispatch mechanism dispatches the new thread to one
4 of the plurality of processors that can accept an additional thread.

- 1 11. (Original) The program product of claim 7 wherein, if all of the plurality of
2 processors cannot accept an additional thread, the thread dispatch mechanism
3 waits for one of the plurality of processors to complete processing a thread,
4 thereby becoming a processor that can accept an additional thread, and then
5 dispatches the thread to the processor that can accept an additional thread.
- 1 12. (Previously added) The apparatus of claim 1 wherein all processors are made
2 busy with a first thread before dispatching a second thread to any processor.
- 1 13. (Previously added) The method of claim 4 wherein all processors are made
2 busy with a first thread before dispatching a second thread to any processor.
- 1 14. (Previously added) The program product of claim 7 wherein all processors are
2 made busy with a first thread before dispatching a second thread to any processor.